Caracterització de locutors usant deep learning

Lliurament Final

Miguel Alcón Doganoc
Director: Javier Hernando Pericas
Ponent: Lluís Padró Cirera

Facultat d'Informàtica de Barcelona
26 de juny de 2018

Els éssers humans som capaços de reconèixer a un altre ésser humà a partir de la seva veu. D'igual manera, som capaços de jutjar si dues persones tenen la mateixa veu o no. Com a futur enginyer informàtic em pregunto: estic capacitat per transmetre aquest coneixement a un ordinador? En aquest treball desenvolupo un sistema informàtic capaç de comparar dos fitxers d'àudio, on parla una persona, i determinar si comparteixen locutor, amb un error òptim del 29,77%. El sistema informàtic consta de dues parts: el nucli computacional i la interfície gràfica. Aquest nucli està format per una xarxa neuronal artificial, un dels mètodes més coneguts del *deep learning*. La interfície, desenvolupada amb HTML, CSS i JavaScript, és l'eina que l'usuari final farà servir per realitzar la comparació de locutors, és a dir, per utilitzar la xarxa.

Los seres humanos somos capaces de reconocer a otro ser humano por su voz. De igual forma, somos capaces de juzgar si dos personas tienen la misma voz o no. Como futuro ingeniero informático me pregunto: ¿estoy capacitado para transmitir este conocimiento a un ordenador? En este trabajo desarrollo un sistema informático capaz de comparar dos archivos de audio, donde habla una persona, y determinar si comparten locutor, con un error óptimo del 29,77%. El sistema informático consta de dos partes: el núcleo computacional y la interfaz gráfica. Este núcleo está formado por una red neuronal artificial, uno de los métodos más conocidos del *deep learning*. La interfaz, desarrollada con HTML, CSS y JavaScript, es la herramienta que el usuario final utilizará para realizar la comparación de locutores, es decir, para usar la red.

We, humans, are able to recognize other humans by their voice. We are also able to judge if two people have the same voice or not. As a future computer engineer, I ask me: am I trained to transmit this knowledge to a personal computer? In this project, I develop a computer system capable of comparing two audio files, where one person talks, and determine if these sequences share speakers, with an optimal error rate of 29,77%. The computer system consists of two parts: the computational nucleus and the graphics interface. This nucleus is composed of an artificial neural network, one of the most famous methods of deep learning. The interface, developed using HTML, CSS, and JavaScript, is the tool that the final user will use to compare two speakers, i.e., to use the network.

Vull agrair a en Javier Hernando Pericas i a en Lluís Padró Cirera per haver-me dirigit en el meu Treball de Final de Grau. A en Miquel Angel India Massana per ajudar-me en la part més tècnica d'aquest. I, per descomptat, a la meva família, per donar-me suport en tot moment i concedir-me l'oportunitat de realitzar-lo.

Índex

Índex de figures

Índex de taules

Capítol 1

Context

1.1 Introducció

El senyal de veu transmet molts tipus d'informació diferent. La principal, i la que ens permet comunicar-nos de forma efectiva, és la informació que transmetem a través de les paraules, però aquesta no és l'única. El senyal de veu també dóna informació sobre la identitat del locutor, és a dir, edat, gènere, idioma, accent, estat emocional i físic, etc. [1] Utilitzar tota aquesta informació per caracteritzar al locutor, té una àmplia gamma d'aplicacions comercials [15], de seguretat [14] i mèdiques, en escenaris del món real [2, 3]. Per aquest motiu, la caracterització de locutors ha estat un dels focus del *machine learning*.

La caracterització de locutors, o caracterització de la parla, és el procés d'identificar automàticament el conjunt de característiques del senyal de veu d'un locutor. Consta de dues etapes fonamentals: transformar el senyal en informació útil que pugui ser tractada i, utilitzant aquesta informació, adquirir les característiques desitjades del locutor. És justament en la realització d'aquestes tasques on entra en joc el *deep learning*.

El *deep learning* forma part d'una gran família de mètodes de *machine learning* basats en l'aprenentatge de representacions de dades. Algun dels seus models estan vagament inspirats en el processament d'informació i en els patrons de comunicació del sistema nerviós biològic, i són coneguts com a xarxes neuronals artificials. Aquestes xares són sistemes informàtics que aprenen a realitzar tasques a partir d'un conjunt d'exemples.

Algunes de les arquitectures del *deep learning* han estat aplicades a camps com el processament del llenguatge natural, la traducció automàtica, el reconeixement de la parla i d'àudio, etc., on s'han aconseguit resultats iguals o superiors als d'humans experts [4]. Per aquesta raó, el *deep learning* és considerat, actualment, un dels mètodes més populars que cal tenir en compte a l'hora de realitzar tasques de *machine learning* en el domini acústic. És per això que s'utilitza en aquest treball.

1.2 Estat de l'Art

1.2.1 Resum històric

Els primers intents que es van realitzar en el camp del reconeixement automàtic de locutors, daten del 1960, una dècada més tard que en el camp del reconeixement de la parla [11]. Ja en aquest punt es van definir dos conceptes claus del camp, els quals encara avui dia es tenen en compte.

- *Speaker identification* i *speaker verification*. Defineix si es pretén identificar al locutor, o verificar que el que parla és un en concret.
- *Text dependent* i *text independent*. Defineix si el diàleg del locutor influirà o no a l'hora d'identificar, o verificar el locutor.

A partir d'aquest moment, els dos camps, junt amb altres també relacionats, han anat evolucionant conjuntament, desenvolupant mètodes cada cop més robustos i precisos. Gràcies a l'avanç tecnològic i a mètodes com *hidden Markov model* (HMM), *Gaussian mixture models* (GMM), *neural networks*, etc. s'ha pogut arribar on estem actualment.

1.2.2 Actualitat

Deep learning

Un dels mètodes més punters en el camp de la caracterització de locutors, i el que s'ha utilitzat per al desenvolupament del treball, és el *deep learning*. De forma tècnica, podem definir-lo com una col·lecció d'algorismes, utilitzats per modelar abstraccions d'alt nivell en dades, a través de l'ús d'arquitectures model. Gràcies als últims avanços en els camps del *software* i del *hardware*, ha augmentat la popularitat d'aquest mètode. Sobretot a partir de l'ús de les *Graphics Processing Units* (*GPUs*), les quals han fet que el *deep learning* sigui superior a qualsevol altre mètode de *machine learning*. Un dels seus algorismes més destacats són les *deep neural networks*, les quals han aconseguit un rendiment excel·lent en els camps de reconeixement de veu i vídeo. A més a més, han demostrat ser superiors, en algun dels camps, als mètodes ja esmentats, com GMM en el reconeixement automàtic de la parla. [12].

Caracterització de locutors

Actualment els dos *softwares* més exitosos en el camp del reconeixement de locutors són els desenvolupats per les empreses AGNITIO i Nuance (tot i que Nuance va adquirir AGNITIO el 2016 [13]). En les pròximes línies s'expliquen aquest parell de *softwares*.

- *AGNITIO's Voice ID* [14] és un *software* utilitzat per organitzacions governamentals (policia, exèrcit, intel·ligència, etc.) en més de 30 països, per prevenir el crim,

identificar criminals i proporcionar proves als tribunals. També l'utilitzen centres de contactes, serveis financers, telecomunicacions i sectors de seguretat empresarial.

El mecanisme de reconeixement de locutor és capaç d'extreure la identitat de la veu (*Voice ID*) per després utilitzar-la per identificar qui està parlant.

- *Nuance Recognizer for Contact Centers* [15] és un *software* d'automatització de trucades, capaç d'oferir una gran experiència de servei al client mentre es millora la taxa de contenció del sistema d'autoservei. Ofereix la màxima precisió de reconeixement de la indústria, ja que fomenta converses naturals i humanes.

1.3 Actors implicats

1.3.1 Desenvolupador

Com es tracta d'un Treball de Final de Grau, jo, el desenvolupador Miguel Alcón Doganoc, sóc la persona encarregada d'investigar, documentar i implementar tot el *software* necessari. A més a més, sóc l'encarregat de gestionar el projecte i de documentar-lo. He d'estar sempre d'acord amb el director i el ponent del treball, i sóc el responsable d'assolir els objectius en les dates establertes.

1.3.2 Director i Ponent

El director, Javier Hernando Pericas, i el ponent, Lluís Padró Cirera, són els encarregats de supervisar el projecte i d'ajudar i guiar al desenvolupador. Individualment, i en trets generals, el director es farà càrrec sobretot de la part més teòrica del projecte (*deep learning*), mentre que el ponent ho farà de la part pràctica (*software*).

1.3.3 Beneficiaris

TALP

El centre de Tecnologies i Aplicacions del Llenguatge i de la Parla (TALP) [6] de la Universitat Politècnica de Catalunya (UPC) utilitzarà aquest projecte com a demostració de la seva tecnologia, per donar visibilitat a la recerca que es realitza.

TextServer

TextServer [5] és una plataforma de TALP i de la UPC, que ofereix serveis d'anàlisi lingüística de textos (*tokenització*, anàlisi morfològica, desambiguació morfosintàctica, anàlisi sintàctica, desambiguació semàntica, anotació de rols semàntics, resolució de la coreferència,

etc.) per a diferents idiomes (anglès, espanyol, català, gallec, francès, italià, portuguès, alemany, asturià, eslovè, rus, etc.). El projecte es volia incloure en la plataforma, on pot ser usat com un servei més o com un mòdul en altres projectes de recerca o transferència de tecnologia. Tot i no haver-se inclòs per falta de temps, segueix sent potencialment útil per al servidor.

Usuaris

Com és potencialment útil pel servidor, també ho ho és pels usuaris de TextServer. Com actualment la plataforma TextServer és gratuïta (tot i que és probable que passi a ser de pagament en un futur), qualsevol persona interessada podrà beneficiar-se del resultat final.

Capítol 2

Formulació del problema

2.1 Motivació

2.1.1 Caracterització de locutors

Els éssers humans són capaços de reconèixer gairebé tots els sorolls que escolten en el seu dia a dia i actuar, si cal, davant d'ells. De la mateixa forma, també són capaços de reconèixer a un altre ésser humà per la seva veu i identificar-lo. El típic cas de 'Sóc jo, obre!' en contestar després de picar a l'intèrfon de casa, és un bon exemple de com els humans no necessiten cap més informació que la mateixa veu per saber que el seu fill o la seva filla vol entrar a casa i no porta claus. Si ells ho poden fer, per què no ho pot fer una màquina?

Aquesta pregunta ja ha estat mig resposta en el capítol anterior. Actualment una màquina seria capaç de fer-ho amb el *software* adequat. Tot i això, ha calgut més de 50 anys d'investigacions i desenvolupament tecnològic per arribar a aquest punt. En aquest projecte s'ha desenvolupat un *software* capaç de comparar dos fitxers d'àudio per determinar si comparteixen locutor.

2.1.2 TextServer

Com ja s'ha explicat a l'apartat 1.3.3, el *TextServer* és una plataforma que ofereix serveis d'anàlisi sintàctica de textos. Tot i això, es vol ampliar perquè també sigui capaç d'analitzar fitxers d'àudio. És per això que part del treball es volia integrar a la plataforma.

2.2 Objectius

El projecte es va dividir en 8 objectius. Els 5 principals garanteixen el desenvolupament del mòdul de verificació de locutors, i el d'una interfície on poder ensenyar el seu funcionament.

Els altres 3 s'encarreguen que el mòdul s'inclogui al *TextServer*. Finalment, per falta de temps, no he pogut realitzar els objectius secundaris.

2.2.1 Principals

1. Adquirir els coneixements necessaris sobre *deep learning* (*deep neural network, siamese neural network,* etc.)
2. Trobar i familiaritzar-me amb una llibreria de C++, sobre *deep learning*, que permeti el desenvolupament del *software*.
3. Desenvolupar el mòdul de verificació de locutors, aplicant els coneixements adquirits als dos punts anteriors.
4. Dissenyar i desenvolupar una interfície gràfica (*HTML, CSS* i *JavaScript*, coneixements que ja tinc) per a la demostració.
5. Fusionar la interfície gràfica amb el mòdul de verificació de locutors per obtenir la demostració final del treball, i la que el centre TALP usarà per donar visibilitat a la seva recerca.

2.2.2 Secundaris

6. Aprendre com crear un servei web a partir d'un *software* desenvolupat en C++, dins una plataforma web.
7. Aplicar els coneixements adquirits a l'apartat anterior i incloure el mòdul al *TextServer* com a servei web.
8. Actualitzar la demostració, utilitzant aquest servei web en lloc del mòdul de forma directa.

Capítol 3

Abast

3.1 Meta final

A la fita inicial vaig definir una meta final pel treball. En aquesta, especificava com havia de ser el resultat de tots els objectius, mencionats al capítol anterior. És justament això el que s'explica a continuació.

El mòdul de verificació de locutors havia de realitzar una comparació entre dos fitxers d'àudio, i determinar si aquestes comparteixen locutors. Aquest mòdul s'havia d'utilitzar per a la creació de la demostració i s'havia d'incloure al *TextServer*.

La integració com a servei web del mòdul a la plataforma *TextServer* havia de permetre que altres projectes de recerca o transferència de tecnologia poguessin utilitzar-lo.

La demostració havia d'utilitzar el mòdul desenvolupat per comparar dos fitxers d'àudio a través d'una interfície gràfica. Una altra opció era que s'utilitzés el servei web en comptes del mateix mòdul per fer funcionar la demostració.

3.2 Possibles obstacles

A part de definir una meta final, també vaig plantejar-me els possibles contratemps amb els quals em podia topar. En les següents línies es plantegen els que s'esperaven, i possibles formes de solucionar algun d'ells.

3.2.1 Errors de programació

És l'obstacle per excel·lència en el desenvolupament d'un *software*, el que qualsevol programador es troba diàriament. Cal tenir cura a l'hora d'escriure el codi, tenir clar el que es vol programar i com es vol fer. A més a més, cal provar que allò que s'ha realitzat funciona correctament abans d'avançar.

3.2.2 Manca de coneixements

El *deep learning* és un mètode complex i no tenia els coneixements necessaris sobre el tema. Havia d'assolir uns nivells mínims per a poder realitzar el projecte de forma satisfactòria. Tot i això, esperava que els meus coneixements previs en *machine learning* agilitzessin l'aprenentatge.

A més a més, el mòdul de verificació de locutors s'ha desenvolupat utilitzant una llibreria de C++, DyNet[26]. Havia de familiaritzar-me amb ella ràpidament.

No he inclòs mai un servei web en una plataforma web.

3.2.3 Mostres per l'entrenament

Una de les etapes més importants de qualsevol mètode de *machine learning* és la d'entrenament. Per a realitzar un bon entrenament calen moltes, i molt diverses, mostres d'entrada que, en el cas d'aquest projecte, són fitxers d'àudio. Podia donar-se la situació que no es disposessin de suficients mostres o que aquestes no fossin suficientment variades.

3.2.4 Calendari

La poca quantitat de temps que es disposa per al desenvolupament del treball, junt amb els altres obstacles esmentats, podia arribar a ser un problema. Havia de ser indispensable una bona planificació de la feina a realitzar i l'assoliment a temps dels objectius.

Capítol 4

Planificació temporal

4.1 Programació

El projecte va començar el 19 de febrer de 2018 amb l'inici de l'assignatura de Gestió de Projectes, i finalitza el 3 de juliol de 2018 amb la lectura del TFG[1], uns 4 mesos després.

4.2 Descripció de les tasques

Tal com s'ha vist a l'apartat 2.2, hi ha diversos objectius a realitzar. Per poder-los complir de forma efectiva, inicialment es van dividir en les següents tasques.

4.2.1 Fita inicial

És l'inici del projecte i correspon a l'aprenentatge i documentació de l'assignatura Gestió de Projectes. Consta de sis lliuraments ben definits:

1. Abast del projecte i contextualització.
2. Planificació temporal
3. Gestió econòmica i sostenibilitat
4. Presentació preliminar
5. Plec de condicions (Especialitat: Computació)
6. Presentació oral i document final

[1]Treball de Fi de Grau.

4.2.2 Mòdul de verificació de locutors

Adquisició dels coneixements necessaris sobre deep learning

És una de les etapes més importants del projecte. Aprenc tot el necessari sobre *deep learning* per triar la llibreria que s'adeqüi millor i, sobretot, per poder desenvolupar el mòdul.

Cerca de la llibreria de C++ adient i familiarització

Busco una llibreria de C++ per ajudar en la realització del mòdul de verificació de locutors. A més a més, em familiaritzo amb ella per ser més eficient a l'hora de desenvolupar-lo.

Desenvolupament

Amb els coneixements previs adquirits, desenvolupo el mòdul de verificació de locutors. Aquesta tasca es pot dividir en 3 parts:

1. Creació de la xarxa neuronal artificial siamesa.
2. Entrenament i prova de la xarxa.
3. Comprovació dels resultats. Si els resultats no són bons, tornar al punt 1.

4.2.3 TextServer

Adquisició dels coneixements web necessaris

Aprenc a crear un servei web, amb el mòdul de verificació de locutors, i a integrar aquest servei a un servidor.

Inclusió del mòdul

Amb els coneixements previs adquirits, incloc el mòdul a la plataforma web *TextServer*.

4.2.4 Demostració

Disseny

Dissenyo la interfície gràfica de la demostració perquè sigui el més usable i agradable a la vista possible. Això m'ajuda a tenir una idea clara abans de posar-me a desenvolupar-la i em facilita el compliment d'aquests dos requisits.

Desenvolupament

Amb el disseny ben clar, desenvolupo la demostració del projecte. Aquesta es realitza amb eines web. No es requereixen coneixements nous sobre aquest tema (*HTML, CSS, JavaScript*), perquè ja els tinc.

4.2.5 Fita final

La fita final és l'última etapa del TFG i consta de tres tasques:

1. Comprovació del funcionament de tot el *software* realitzat.
2. Redacció de la memòria del projecte.
3. Preparació per la lectura final.

4.3 Temps total

4.3.1 Estimació inicial

Tasca	Duració aproximada (h)
Fita inicial	90
Mòdul de verificació de locutors	180
TextServer	60
Demostració	50
Fita final	100
Total	480

Taula 4.1: Temps estimat dedicat al desenvolupament de cada tasca.

4.3.2 Realitat

Tasca	Duració aproximada (h)
Fita inicial	90
Mòdul de verificació de locutors	320
TextServer	0
Demostració	60
Fita final	80
Total	550

Taula 4.2: Temps real destinat al desenvolupament de cada tasca.

4.4 Diagrama de Gantt

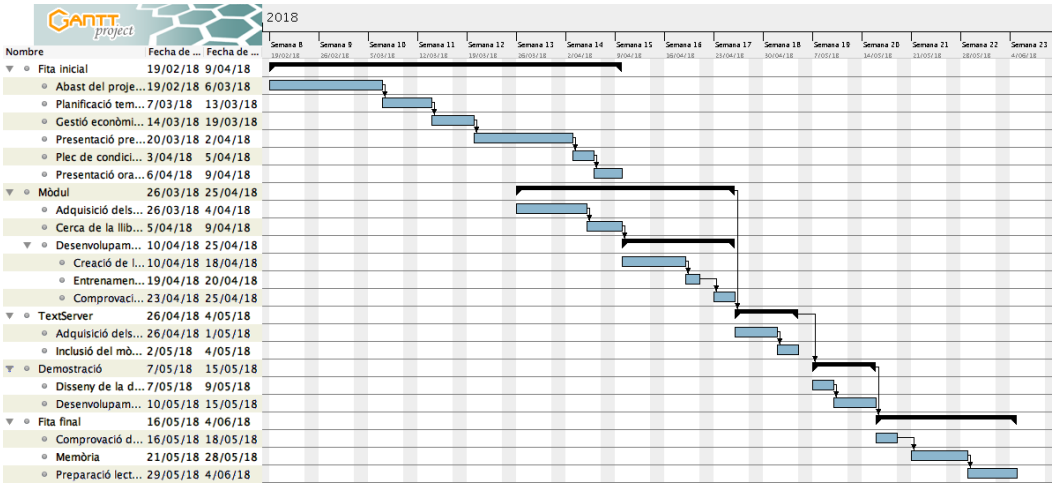


Figura 4.1: Diagrama de Gantt inicial

Totes les tasques havien de ser, i han estat, realitzades per mi. Per aquest motiu, totes les dependències del diagrama inicial (figura 4.1) són d'una tasca amb la seva predecessora, excepte la referent al mòdul de verificació de locutors. Aquesta havia de començar abans d'acabar la fita inicial per poder complir amb les dates previstes, i perquè la càrrega de treball del moment era baixa.

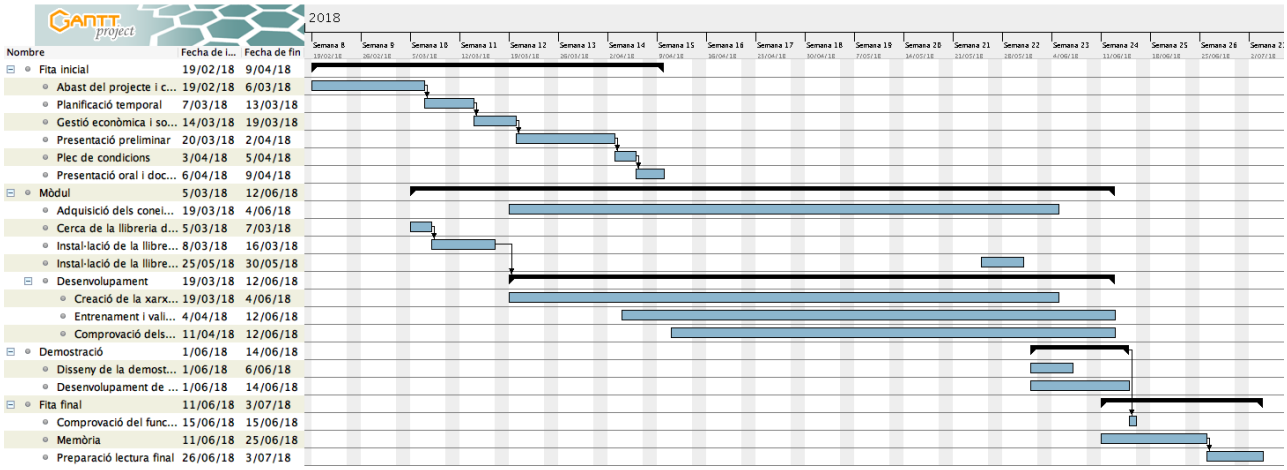


Figura 4.2: Diagrama de Gantt final

Aquesta distribució inicial del temps no ha acabat sent gaire precisa. Com es pot apreciar al diagrama final (figura 4.2), gran part de les dependències ja no estan. Això es deu al fet que la majoria de tasques es poden començar abans d'acabar l'anterior i, quan m'he topat amb algun problema, he aprofitat el temps amb una altra tasca. Tot i això, sí que hi ha restriccions entre alguna de les tasques.

- Es pot començar amb l'entrenament de la xarxa un cop aquesta hagi estat creada.
- Es pot començar amb la comprovació dels resultats un cop la xarxa estigui entrenada.
- La demostració necessita que el mòdul estigui desenvolupat del tot.
- La fita final no pot acabar sense finalitzar amb la resta del treball.

La resta de canvis realitzats en la planificació s'expliquen més endavant, a l'apartat 4.6.

4.5 Imprevists

Com ja he comentat a l'aparat 3.2.2, per desenvolupar el mòdul de verificació de locutors he utilitzat la llibreria *DyNet*. Aquesta llibreria no és una de les més utilitzades, no hi ha gaire informació fora de la web oficial. Per aquest motiu, qualsevol problema amb el qual m'he topat, ha estat més difícil trobar una solució.

El primer problema que em vaig trobar, i que no havia tingut en compte a la planificació inicial, és el de la seva instal·lació. Em va portar uns dies fer-la funcionar al meu ordinador personal. Un cop realitzat això, em vaig adonar que el meu ordinador no tenia suficient potència per entrenar la xarxa neuronal en un temps raonable, i els directors em van proporcionar un compte a la xarxa d'ordinadors *Calcula*[2], la qual compta amb diverses màquines amb molta memòria, *CPUs* i *GPUs*. Instal·lar *DyNet* a *Calcula* ens va portar temps, perquè té diverses dependències i no disposàvem amb el compte de superusuari.

El segon problema ha estat que el temps requerit per adquirir els coneixements necessaris per desenvolupar el mòdul, i el mateix desenvolupament, ha estat superior a l'esperat.

El tercer i últim problema se'ns va presentar unes setmanes abans d'acabar-lo. L'ús de *CPUs* no era suficient per poder analitzar l'entrenament de la xarxa i actuar ràpidament en conseqüència. Vam decidir utilitzar les *GPUs* però *DyNet* requereix una instal·lació especial per poder-les utilitzar. Aquesta instal·lació ens va portar uns dies més.

4.6 Canvis realitzats

La planificació ha patit diverses modificacions al llarg del treball. A causa dels imprevists esmentats a l'apartat anterior, la realització de les tasques s'ha vist afectada de la següent manera:

- El temps requerit per al desenvolupament del mòdul de verificació de locutors ha augmentat dràsticament, com a conseqüència de les dues instal·lacions de *DyNet* i de la dificultat de la tasca.
- La inclusió del mòdul al *TextServer* no s'ha realitzat.

[2]Calcula és l'arquitectura que s'utilitza al departament de Teoria del Senyal i Comunicacions de la UPC per administrar el conjunt de màquines i serveis del departament.

A part dels canvis causats pels imprevists, el mateix desenvolupament del treball m'ha fet alterar el transcurs d'alguna de les tasques.

- L'obtenció dels fitxers d'àudio, mencionat com un possible problema a l'apartat 3.2.3, no l'he dut a terme perquè els directors ja s'han encarregat de proporcionar-m'ho abans de començar amb l'entrenament de la xarxa.
- El temps dedicat a l'aprenentatge ha conviscut amb el del desenvolupament. Aprendre la part més teòrica juntament amb la part pràctica, ha estat la forma més natural de treballar.
- El mateix ha passat amb la comprovació dels resultats. Un cop la xarxa ha estat creada, entrenada i validada, els mateixos resultats demanaven modificar-la, entrenar-la i validar-la. Per aquest motiu, aquestes 3 tasques també han conviscut en el temps.
- El disseny i el desenvolupament de la demostració també han acabat convivint perquè, un cop desenvolupada, ha patit diverses millores visuals.

A més a més, les dates d'entrega i de lectura del treball s'han endarrerit una setmana.

Capítol 5

Recursos

5.1 Recursos personals

L'únic recurs personal del qual s'ha disposat per realitzar aquest projecte és el del desenvolupador.

El temps de dedicació ha estat d'unes 28 hores setmanals, les necessàries per complir amb el diagrama de Gantt (figura 4.2) i amb el temps estimat (taula 4.2).

5.2 Recursos materials

5.2.1 Hardware

- *MacBook Pro 2016 13"* (sense *Touch Bar*) [16]. Eina *hardware* amb la qual s'ha desenvolupat tot el *software* del projecte, i amb el que s'ha escrit tota la documentació.
- *Calcula*. Xarxa d'ordinadors que s'ha utilitzat per entrenar la xarxa neuronal siamesa.

5.2.2 Software

- *macOS High Sierra* [17]. Sistema operatiu amb el qual s'ha treballat durant tot el projecte.
- *Ubuntu 18.04 LTS* [18]. Sistema operatiu del servidor *Calcula*.
- *Virtual Studio Code* [19]. Eina *software* d'edició de texts. S'ha utilitzat per escriure el codi del *software* i tota la documentació.
- *Electron* [20]. Eina *software* (*framework*) que s'ha utilitzat per desenvolupar la demostració. Permet crear aplicacions multiplataforma d'escriptori utilitzant *HTML, CSS i JavaScript*.

- *Git*[21] i *GitHub* [22]. Són, respectivament, una eina *software* per al control de versions del projecte i un repositori on es guarda el projecte.
- LaTeX [23]. Eina *software* que s'ha utilitzat per a la redacció de la documentació del treball.
- *GanttProject* [24]. Eina *software* que s'ha utilitzat per fer els diagrames de Gantt.
- *SoftCatalà: Corrector ortogràfic i gramatical* [25]. Eina *software* que s'ha utilitzat per detectar i corregir errors ortogràfics i gramaticals en la documentació.

5.3 Recursos destinats a despeses generals

- Electricitat. Necessària per alimentar el meu ordinador, amb el que s'ha desenvolupat tot el projecte, i la xarxa d'ordinadors *Calcula*, amb la qual s'ha entrenat la xarxa.
- Fibra òptica. Necessària per buscar informació, consultar documents, descarregar la llibreria, etc.
- Transport. Necessari per assistir a les reunions.

Capítol 6

Gestió econòmica

6.1 Identificació dels costos

Per poder dur a terme aquest projecte, s'han necessitat tots els recursos esmentats al capítol 5. Aquests recursos són els que han generat tots els costos del treball. Al llarg d'aquest capítol, mostro el pressupost inicial, que vaig realitzar a la fita inicial, i el comparo amb el que ha acabat sent.

6.2 Estimació dels costos

6.2.1 Pressupost d'eines hardware

Per poder dur a terme totes les tasques, s'ha utilitzat el meu ordinador portàtil. A la taula 6.1 es mostra l'estimació del cost d'aquest, tenint en compte la seva vida útil i les seves amortitzacions.

Producte	Preu	Unitats	Vida útil	Amortització
MacBook Pro 2016 13"	1.505,59 €	1	5 anys	100,37 €

Taula 6.1: Pressupost inicial d'eines hardware.

Com he mencionat a l'apartat 4.5, per entrenar la xarxa neuronal he utilitzat la xarxa d'ordinadors *Calcula*. A la fita inicial encara no sabia que la utilitzaria i, per tant, no la vaig tenir en compte. Com *Calcula* és propietat de la UPC, i no he pagat res per utilitzar-la, he calculat el preu com si hagués entrenat la xarxa neuronal a la *Google Cloud Platform*, utilitzant la seva calculadora de preus [28].

Producte	Preu	Unitats	Vida útil	Amortització
MacBook Pro 2016 13”	1.505,59 €	1	5 anys	100,37 €
Calcula	-	-	-	20,81 €
Total				121,18 €

Taula 6.2: Cost final d’eines hardware.

6.2.2 Pressupost d’eines software

A part del *hardware*, s’han necessitat també eines *software* per poder desenvolupar el projecte. Com totes les que he utilitzat són totalment gratuïtes, el cost total del *software* és zero. Per aquest motiu, les amortitzacions també tindran cost zero.

Producte	Preu	Unitats	Vida útil	Amortització
macOS High Sierra	0 €	1	-	0 €
Virtual Studio Code	0 €	1	-	0 €
Git	0 €	1	-	0 €
GitHub	0 €	1	-	0 €
LaTeX	0 €	1	-	0 €
GanttProject	0 €	1	-	0 €
SoftCatalà	0 €	1	-	0 €
Total	0 €			0 €

Taula 6.3: Pressupost inicial d’eines software.

Producte	Preu	Unitats	Vida útil	Amortització
macOS High Sierra	0 €	1	-	0 €
Ubuntu 18.04 LTS	0 €	1	-	0 €
Virtual Studio Code	0 €	1	-	0 €
Electron	0 €	1	-	0 €
Git	0 €	1	-	0 €
GitHub	0 €	1	-	0 €
LaTeX	0 €	1	-	0 €
GanttProject	0 €	1	-	0 €
SoftCatalà	0 €	1	-	0 €
Total	0 €			0 €

Taula 6.4: Cost final d’eines software.

6.2.3 Pressupost de recursos humans

Com ja s’ha comentat a l’apartat 1.3.1, el projecte l’ha dut a terme només una persona. He hagut d’assolir el rol de cap de projecte (CP), desenvolupador de *software* (DS), dissenyador

de *UI/UX (user interface/user experience)* (DIX) i *tester* (T), durant les 550 h de duració del treball, tal com es va aproximar al capítol 4.

A la taula 6.5 es troba l'estimació dels costos dels recursos humans, segons el rol i les hores que treballa cadascun d'ells. La repartició d'aquestes hores, segons la tasca a realitzar (cada etapa del diagrama de Gantt), es troba a la taula 6.6.

Rol	Hores	€/hora	Salari
Cap de projecte	180	50	9.000 €
Desenvolupador de software	240	35	8.400 €
Dissenyador UI/UX	20	35	700 €
Tester	40	30	1.200 €
Total	480		19.300 €

Taula 6.5: Pressupost inicial de recursos humans.

Tasca	Duració (h)	Dedicació (h)			
		CP	DS	DIX	T
Fita inicial	90	90	0	0	0
Mòdul de verificació de locutors	180	20	150	0	10
TextServer	60	0	60	0	0
Demostració	50	0	30	20	0
Fita final	100	70	0	0	30
Total	480	180	240	20	40

Taula 6.6: Estimació inicial de temps per tasca i rol.

En les dues taules següents, es mostren la repartició d'hores segons la tasca a realitzar i el rol, i el cost final dels recursos humans, després dels canvis realitzats a la planificació temporal inicial (apartat 4.6).

Tasca	Duració (h)	Dedicació (h)			
		CP	DS	DIX	T
Fita inicial	90	90	0	0	0
Mòdul de verificació de locutors	320	40	240	0	40
TextServer	0	0	0	0	0
Demostració	60	0	45	10	5
Fita final	80	70	0	0	10
Total	550	200	285	10	55

Taula 6.7: Temps final per tasca i rol.

Rol	Hores	€/hora	Salari
Cap de projecte	200	50	10.000 €
Desenvolupador de software	285	35	9.975 €
Dissenyador UI/UX	10	35	350 €
Tester	55	30	1.650 €
Total	550		21.975 €

Taula 6.8: Cost final de recursos humans.

6.2.4 Costos indirectes

A part de tots els costos directes del projecte, mostrats als apartats anteriors, també s'han utilitzat altres recursos com electricitat, connexió a internet i transport públic, explicats a l'apartat 5.3.

Producte	Preu	Unitats	Cost estimat
Electricitat	0,14 €/kWh	480 kWh	67,2 €
Fibra òptica	40 €/mes	4 mesos	160 €
Transport	1,02 €/viatge	30 viatges	30,6 €
Total			257,8 €

Taula 6.9: Estimació inicial dels costos indirectes.

Als costos indirectes també he afegit l'electricitat gastada per *Calcula*. He calculat el seu consum a partir de les especificacions de la *GPU* utilitzada (*NVIDIA TITAN Xp* [29]).

Producte	Preu	Unitats	Cost estimat
Electricitat PC	0,14 €/kWh	480 kWh	67,2 €
Electricitat *Calcula*	0,14 €/kWh	250 kWh	35 €
Fibra òptica	40 €/mes	4 mesos	160 €
Transport	1,02 €/viatge	30 viatges	30,6 €
Total			292,8 €

Taula 6.10: Costos indirectes finals.

6.2.5 Costos inesperats

Com era possible que sorgissin diversos imprevists durant el desenvolupament del projecte, tal com ha acabat passant (apartat 4.5), a la fita inicial vaig decidir deixar un marge del 10% del pressupost total (contingència) per aquests possibles contratemps. A la taula 6.11 es pot veure que aquest marge és de 1.965,81 €.

6.2.6 Pressupost total

Tenint en compte tots els costos inicials definits en aquest capítol, es va obtenir un pressupost total de 21.623,98 € (taula 6.11). Tot i haver deixat un marge de contingència per a imprevists, no ha estat suficient per cobrir els 22.388,98 € del cost total (taula 6.12). El pressupost inicial i el cost final difereixen en 765 €.

Motiu	Cost estimat
Hardware	100,37 €
Software	0 €
Recursos humans	19.300 €
Costos indirectes	257,8 €
Subtotal	19.658,17 €
Contingència (10 %)	1.965,81 €
Total	21.623,98€

Taula 6.11: Pressupost total inicial.

Motiu	Cost
Hardware	121,18 €
Software	0 €
Recursos humans	21.975 €
Costos indirectes	292,8 €
Total	22.388,98 €

Taula 6.12: Cost total del projecte.

6.3 Viabilitat

Després de presentar tots els costos del projecte, vaig haver d'avaluar si aquest era viable o no. Encara que el pressupost inicial és de 21.623,98 €, cal no oblidar-se que excepte els recursos *hardware* i indirectes (358,17 €), la resta de costos són els de recursos humans. És a dir, tots aquests diners anirien destinats a les persones que desenvolupessin el projecte, que en aquest cas sóc jo, l'autor del treball.

Per tant, aquest projecte és econòmicament viable.

Capítol 7

Metodologia i rigor

7.1 Mètode de treball

En el desenvolupament del treball s'ha utilitzat una metodologia de prototipatge [7], on un prototip[1] es desenvolupa, es prova, i es refà fins que finalment s'aconsegueix arribar al producte final, o a un prototip acceptable, a partir del qual es pot començar a desenvolupar el producte sencer. El primer prototip sol ser una aproximació llunyana del producte final, però a cada etapa del procés es va ampliant i millorant. En el meu cas, aquest prototip ha estat un dels exemples de *DyNet* [27], el qual utilitza una xarxa neuronal per reconèixer números manuscrits.

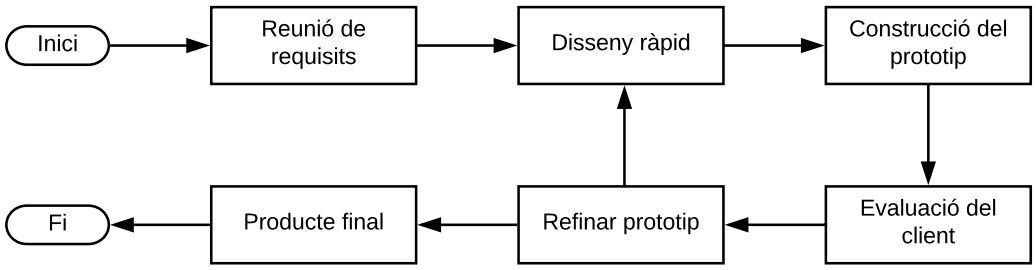


Figura 7.1: Diagrama de la metodologia de prototipatge [8].

Aquesta metodologia funciona millor quan no es coneixen al detall tots els requeriments del projecte abans de començar-lo. Segueix un procés iteratiu, de prova i error, que se sol realitzar entre desenvolupadors i usuaris.

[1]Un prototip és una aproximació del producte final.

7.2 Eines de seguiment

Per a la realització del *software* s'ha utilitzat Git, el famós sistema de control de versions gratuït i de codi obert, junt amb GitHub, per disposar d'un repositori[2] en línia al que poder accedir des de diferents dispositius. Amb aquestes dues eines s'ha pogut realitzar un millor seguiment del procés del projecte (documentació de tots els canvis realitzats, errors, etc.) i s'han pogut consultar totes les versions enregistrades.

Per a la comunicació amb el director i el ponent s'ha utilitzat el correu electrònic.

7.3 Mètode de validació

A la fita inicial vaig plantejar-me com validaria els resultats obtinguts. A continuació explico el que creia que faria, i al capítol 8 es detalla el que s'ha realitzat finalment.

7.3.1 Mòdul de verificació del locutor

Utilitzant fitxers d'àudio que no s'hagin fet servir a l'entrenament, es provarà l'eficàcia del mòdul comparant-los. Segons els resultats obtinguts, s'haurà de millorar el mòdul o es podrà seguir avançant amb el projecte.

7.3.2 Integració del mòdul al *TextServer*

Un cop estigui integrat el mòdul, es provarà d'accedir a ell a través del *TextServer*. Si s'aconsegueix, es donarà la tasca per validada.

7.3.3 Demostració del mòdul

Es realitzaran les mateixes proves que amb el mòdul, però en aquest cas a través de la demostració. A més a més, es comprovarà el funcionament de la interfície, assegurant que es comporti com s'espera.

[2]El meu repositori es troba a `https://github.com/miquelalcon/speaker-recognition`.

Capítol 8

Desenvolupament

8.1 Mòdul de verificació de locutors

El mòdul de verificació de locutors és un programa escrit en C++ utilitzant la llibreria DyNet. Com ja s'ha explicat en capítols anteriors, aquest programa és capaç de comparar si dos fitxers d'àudio corresponen al mateix locutor o a dos de diferents. Utilitza una xarxa neuronal siamesa per a realitzar aquesta decisió.

8.1.1 Preprocessament dels fitxers d'àudio

Abans de tractar el preprocessament, cal detallar informació sobre els fitxers d'àudio utilitzats. Aquests fitxers provenen d'una base de dades de converses telefòniques, entre dues persones, en anglès. Abans de preprocessar-los, s'han extret els senyals de cada veu per obtenir un fitxer per locutor. Això és possible perquè els fitxers són estereofònics, és a dir, cada locutor ha estat enregistrat a través d'un canal d'entrada diferent, i es poden separar fàcilment.

Pel que fa al preprocessament dels fitxers d'àudio, està fora de l'objectiu d'aquest projecte i, per aquest motiu, se m'han entregat els arxius corresponents a aquest preprocessament, perquè els utilitzi directament a l'entrenament, a la validació i a la prova de la xarxa. Tot i això, a continuació explico, sense entrar en detalls, com s'ha realitzat aquest procés.

Primer de tot, s'extreuen les característiques, conegudes com a *Frequency Filtering features* [30], de cadascun dels fitxers d'àudio d'entrenament. Aquestes característiques s'extreuen cada 10 ms utilitzant una finestra de 30 ms. El resultat d'aquesta extracció és un conjunt de vectors de característiques de 33 dimensions per cada fitxer. Després, utilitzant aquests conjunts de vectors, es crea un *Universal Background Model* (UBM) [31], és a dir, un model que serveix per representar característiques generals i independents a la persona. Està format per un GMM[1] de 512 components i s'utilitza per generar els super-vectors que

[1]Un *Gaussian Mixture Model* és un model probabilístic que representa la presència de subpoblacions dins d'una població global.

representen a cada locutor. Aquests super-vectors són els arxius que he rebut per entrenar, validar i provar la xarxa.

8.1.2 Xarxa neuronal siamesa

Teoria

Una xarxa neuronal artificial és un sistema computacional capaç d'aprendre a realitzar tasques a partir d'exemples, sense necessitat d'especificar cap norma. Per exemple, si utilitzes una xarxa per reconèixer ocells, no li especifiques que un ocell té plomes, ales, bec, etc. sinó que a partir d'exemples (representació computable d'una imatge), etiquetats amb 'ocell' i 'no ocell', la xarxa aprèn a distingir-los, generant automàticament característiques identificadores a partir d'aquests exemples.

Les xarxes neuronals artificials estan formades per una concatenació de capes. A la vegada, aquestes capes estan formades per una col·lecció de nodes, anomenats neurones artificials. Les neurones d'una capa estan connectades amb totes les neurones de les capes posterior i anterior, si en tenen. Aquestes reben senyals de les neurones de la capa anterior, o de l'entrada, les processen, i envien el senyal resultant a les neurones de la capa posterior, o a la sortida. Els senyals solen ser nombres reals, i el seu processament sol ser una funció no lineal de la suma dels senyals de la capa anterior, o de les dades d'entrada.

Les connexions entre neurones tenen assignades un pes que es va ajustant a mesura que avança l'entrenament. El senyal que rep una neurona d'una altra, depèn d'aquest pes. Sent n_0 una neurona que rep el senyal de les N neurones n_i de la capa anterior, a través de les seves connexions amb pes ω_{i0}, n_0 processa els senyals d'entrada de la següent forma:

$$n_0 = f(\sum_{i=1}^{N} \omega_{i0} n_i + b_0) \tag{8.1}$$

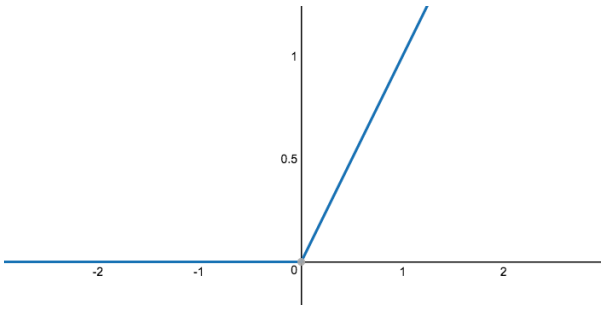
On b_0 és un nombre real arrelat a cada neurona i f és la funció d'activació de la capa de n_0. En aquest treball he utilitzat dues funcions diferents.

- *Rectified Linear Unit* (ReLU):

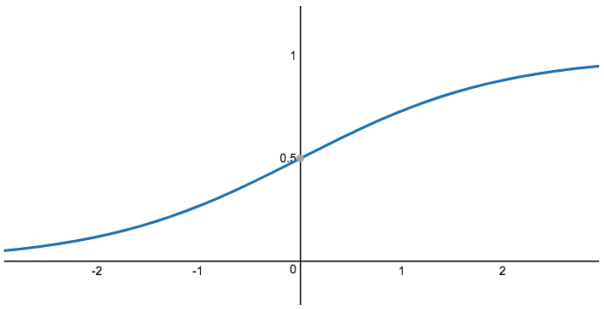
$$f(x) = max(0, x) \tag{8.2}$$

- *Sigmoid*:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{8.3}$$



(a) ReLU function.



(b) Sigmoid function.

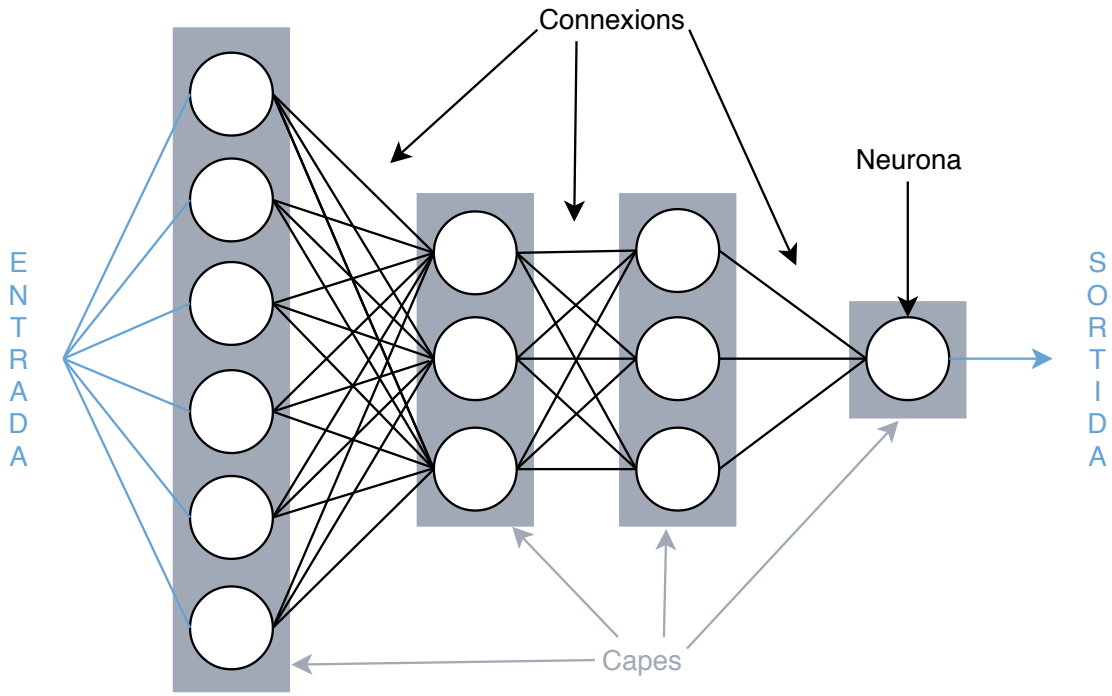


Figura 8.2: Representació gràfica d'una xarxa neuronal senzilla.

L'entrenament d'una xarxa neuronal artificial es basa en la minimització o maximització d'una funció de cost (*loss*). En el cas del meu treball, es vol minimitzar la funció *binary cross entropy*.

$$\mathcal{L}(\omega) = -[y' \log(y) + (1 - y') \log(1 - y')] \tag{8.4}$$

On y és la sortida de la xarxa, configurada amb els pesos ω, davant d'una entrada; y' és la sortida que s'espera (les etiquetes de les quals parlo a l'inici de l'apartat). Per tant, l'entrenament de la xarxa neuronal es basa en obtenir un conjunt de pesos òptims, que minimitzin aquesta funció. Aquest entrenament segueix les següents etapes:

1. *Forward propagation*. S'introdueixen les dades a la xarxa i s'obté una sortida.
2. Es calcula l'error comès (funció *loss*) amb la sortida obtinguda i l'esperada.

3. *Backward propagation.* Es propaga l'error per tota la xarxa i es modifica el pes de cada connexió segons la contribució que ha aportat a l'error.

Sent ω_{ij} el pes de la connexió entre les neurones i i j, t l'etapa actual de l'entrenament, $\frac{\delta\mathcal{L}}{\delta\omega_{ij}}$ la contribució de ω_{ij} a l'error total, λ el *learning rate* i k el *weight decay*, el pes d'una connexió es calcula seguint la formula següent:

$$\omega_{ij}(t+1) = \omega_{ij}(t) - \lambda\frac{\delta\mathcal{L}}{\delta\omega_{ij}} - \lambda k\omega_{ij}(t) \tag{8.5}$$

Perquè l'entrenament de la xarxa es realitzi adequadament, s'han hagut de configurar una sèrie de paràmetres. Aquests paràmetres controlen l'actualització dels pesos de la xarxa i eviten que es produeixi un sobreentrenament[2], és a dir, que la xarxa memoritzi els exemples en lloc d'aprendre les seves característiques.

- *Learning rate.* Paràmetre λ de l'equació 8.5. És la importància que es dóna a la contribució total de l'error, o dit amb altres paraules, la velocitat a la qual la xarxa aprèn. Un *learning rate* mal escollit pot provocar el sobreentrenament de la xarxa.
- *Weight decay.* Paràmetre k de l'equació 8.5. És una tècnica regularitzadora que evita que els pesos de la xarxa siguin grans. En limitar la seva mida, es limita el conjunt de possibles solucions del sistema, i la xarxa és menys propensa al sobreentrenament.
- *Dropout rate.* També és una tècnica regularitzadora de la xarxa. Representa el percentatge de connexions que es desactiven a cada iteració de l'entrenament. D'aquesta manera no se li permet, a la xarxa, fer ús de la totalitat dels seus pesos.

Una xarxa neuronal siamesa és un tipus de xarxa neuronal artificial què conté dues subxarxes idèntiques, és a dir, que tenen la mateixa configuració amb els mateixos pesos. L'actualització dels paràmetres es realitza de la mateixa manera en les dues subxarxes. Aquest tipus de xarxes neuronals són populars a l'hora de trobar similituds o relacions entre dues coses comparables. Quan aquestes realitzen aquest tipus de tasques, solen ser formades per dues subxarxes idèntiques utilitzades per processar dues entrades, i per un mòdul que s'encarrega d'agafar les dues sortides i produir la predicció final.

Aquest tipus de xarxes són bones realitzant tasques de comparació per dos motius:

1. La compartició de pesos representa menys paràmetres a entrenar, és a dir, es requereixen menys dades i menys temps a l'entrenament, i la xarxa té menys tendència a sobreentrenar.
2. Cada subxarxa produeix una representació del seu senyal d'entrada. Si les dues entrades són del mateix tipus, és probable que les subxarxes utilitzin un model similar per processar-les, la qual cosa fa que siguin més fàcils de comparar.

[2]Tot i que 'sobreentrenament' no es troba al diccionari català, l'utilitzo com a traducció de la paraula anglesa *overfitting*.

Construcció

La xarxa utilitzada al mòdul de verificació de locutors està formada per 3 subxarxes. Dues d'elles són les que s'encarreguen, com he comentat a l'apartat anterior, de crear la representació dels senyals d'entrada, és a dir, són les subxarxes siameses. Aquestes estan formades per:

Tipus de Capa	#Neurones	#Connexions	f d'activació	Dropout (%)
Entrada	16896	5000	ReLU	7,5
Intermitja	5000	5000	ReLU	7,5
Intermitja	5000	1000	ReLU	7,5
Sortida	1000	5000	ReLU	7,5

Taula 8.1: Informació sobre les capes que formen les subxarxes siameses.

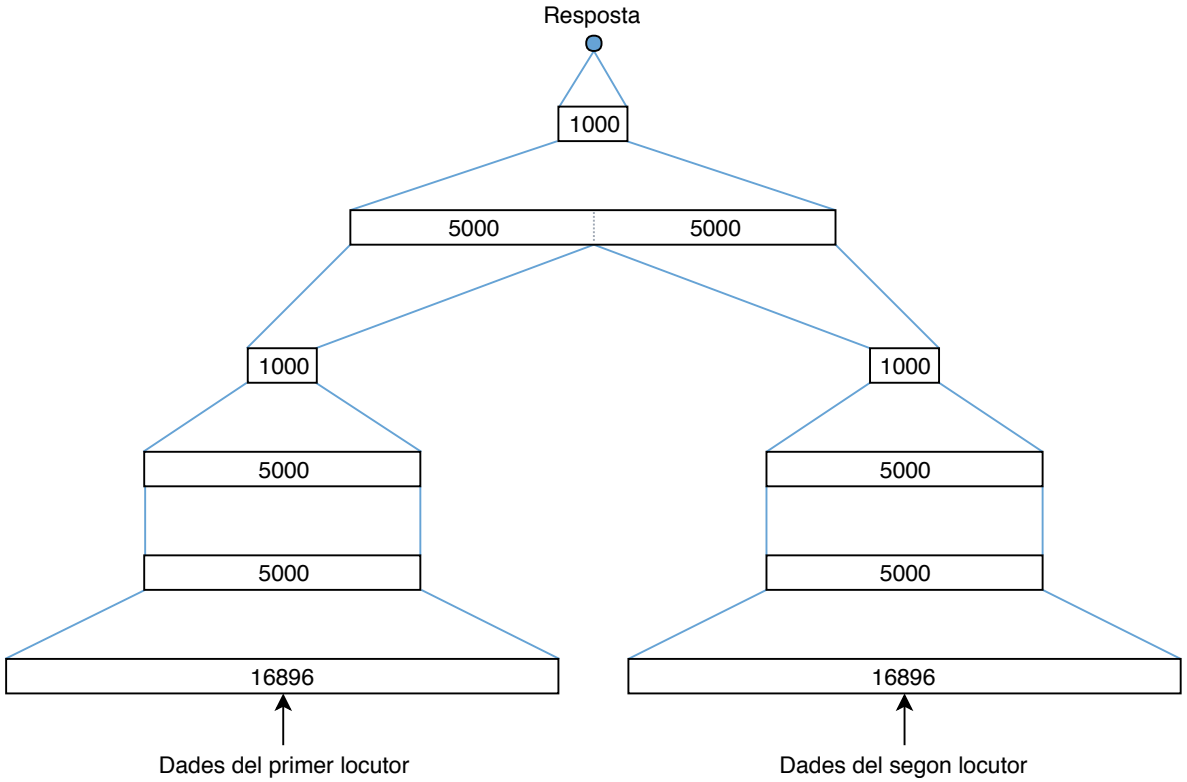


Figura 8.3: Representació gràfica de la xarxa neuronal siamesa utilitzada.

L'altra subxarxa restant s'encarrega de combinar els dos senyals de sortida de les subxarxes siameses i produir una predicció. L'anomenaré subxarxa d'unió, i està formada per:

Tipus de Capa	#Neurones	#Connexions	f d'activació	Dropout (%)
Entrada	10000	1000	ReLU	7,5
Sortida	1000	1	Sigmoid	0

Taula 8.2: Informació sobre les capes que formen la subxarxa d'unió.

Encara que de forma teòrica s'utilitzen 3 subxarxes per formar la xarxa principal, a la pràctica només n'utilitzem dues: una de les siameses i la d'unió.

Divisió de les dades

Les dades utilitzades en l'entrenament i en la validació de la xarxa estan formades per un llistat de locutors, seguit dels seus fitxers d'àudio. Aquestes dades han estat dividides en dues parts.

- Dades per l'entrenament. Sol representar un gran percentatge de les dades, ja que se'n necessiten moltes per poder entrenar, de forma efectiva, una xarxa neuronal. En el cas de les nostres dades, s'ha utilitzat un 80% dels locutors que tenien més de 8 fitxers d'àudio (més de 1000).
- Dades per la validació. Sol representar un percentatge menor de les dades, ja que només les volem utilitzar per veure com es comporta la xarxa davant d'exemples que no ha vist mai. En el cas de les nostres dades, s'ha utilitzat un 20% dels locutors que tenien més de 8 fitxers d'àudio (gairebé 300).

Aquesta divisió es realitza sempre per poder comprovar si la xarxa està realment aprenent (detecta patrons i característiques de les dades) o sobreentrena. Això es pot detectar utilitzant els percentatges d'encert[3] resultants de l'entrenament i de la validació, és a dir, utilitzant la divisió de les dades, es prova la xarxa amb cadascuna de les divisions, i s'anota quants cops encerta. Si el percentatge d'encert en l'entrenament és molt més gran que en la de validació, voldrà dir que la xarxa està sobreentrenant. Si no és així, llavors la xarxa està aprenent correctament.

A part d'aquestes dades d'entrenament i validació, també disposo d'un altre conjunt de dades que s'utilitza en la demostració i per posar a prova la xarxa, les quals anomeno dades de prova.

Entrenament i validació

L'entrenament[4] es realitza creant aleatòriament, i per igual, casos positius i negatius. Aquests exemples es generen d'aquesta manera:

1. Es seleccionen dos locutors diferents de forma aleatòria (primer i segon locutor).

[3] Percentatge d'encert $= \frac{100 \times (\text{nre. d'encerts})}{(nre.dedades)}$.

[4] El codi d'entrenament i validació es troba a l'arxiu `neuralnetwork/siamese_network.cc` del repositori.

2. Es seleccionen aleatòriament dos fitxers d'un dels locutors, creant així un exemple positiu.
3. Es selecciona aleatòriament un fitxer de cada locutor, creant així un exemple negatiu.

Aquesta generació[5] es repeteix 20 cops fins obtenir 40 exemples, els quals s'anomenen *batch*. Cada *batch* es pot dividir en 40 primers locutors, 40 segons locutors i 40 etiquetes[6]. L'entrenament segueix les següents passes:

1. Es genera el *batch*.
2. *Forward propagation* (part 1). S'introdueixen els 40 primers locutors a la subxarxa siamesa, produint així 40 representacions d'aquests.
3. S'introdueixen també els 40 segons locutors a la subxarxa siamesa, produint així 40 representacions d'aquests.
4. Es concatenen les 80 representacions, unint cada representació d'un locutor amb la seva parella de l'exemple.
5. *Forward propagation* (part 2). S'introdueixen les 40 concatenacions de les representacions a la subxarxa d'unió i es genera una sortida[7].
6. Es calcula la mitjana de la funció *loss* amb les prediccions obtinguda i esperades.
7. *Backward propagation*.

Es processen 2500 *batches* abans de validar la xarxa.

La validació de la xarxa és igual que l'entrenament fins al punt 5, excepte perquè utilitza les dades de validació. Després d'aquest punt continua així:

6. Si la predicció és superior o igual a cert llindar (en aquest cas, 0,5), es considera com a cas positiu, sinó com a cas negatiu.
7. S'observa si el cas resultant i l'etiqueta són els mateixos.

Això es realitza 500 cops. Després es calcula el percentatge d'encert.

Tot aquest procés es repeteix com a màxim 1000 cops, i a cada repetició se l'anomena *epoch*. L'algorisme acaba quan s'arriben a les 1000 *epochs*, o si es realitzen més de 10 sense obtenir un percentatge d'encert millor. Al final de tot, es guarda la configuració de la xarxa que ha obtingut el millor percentatge d'encert.

L'entrenament final s'ha realitzat amb un *learning rate* de 0,00001 i un *weight decay* de 0,000001, escollits a través de diversos experiments. El percentatge d'encert obtingut en aquest procés és del 84,985%.

[5]El codi de la generació aleatòria d'exemples es troba a l'arxiu `neuralnetwork/utils.h` del repositori.

[6]L'etiqueta és 1 quan l'exemple és positiu (mateixos locutors) i 0 quan és negatiu (diferents locutors).

[7]La sortida és un nombre real entre 0 i 1. Intuïtivament, com més a prop estigui del 0, més probable que es tracti d'un cas negatiu, i com més a prop estigui de l'1, més probable que es tracti d'un cas positiu.

8.1.3 Resultats

En acabar l'entrenament, cal posar a prova la xarxa, mesurant la seva fiabilitat. Aquestes proves han estat realitzades com si la xarxa fos la base d'un sistema biomètric[8] de verificació de locutor. Utilitzant les dades de prova, he generat un llistat de 2350 clients seguit d'un dels seus fitxers d'àudio. Aquest llistat representa el total de les persones que estan autoritzades per accedir al sistema. Per simular els possibles accessos al sistema, he generat dues llistes més: una pels accessos dels clients, i l'altre pels de persones no autoritzades que volen accedir-hi, els impostors. Estan formades per l'identificador del client, a través del qual es vol accedir, i el fitxer d'àudio de la persona que realitza l'accés. Els llistats utilitzats per provar la xarxa consten de 700 possibles accessos autoritzats i gairebé 30.000 de no autoritzats.

Tot aquest procés[9], traduït al funcionament de la meva xarxa, es basa en comparacions dels fitxers d'àudio de les persones que volen accedir al sistema, amb els arxius corresponents als clients amb els quals es volen realitzar els accessos. Com a resultat d'aquestes comparacions, obtenim les prediccions (nombres reals) de cadascuna d'elles, sense comparar-los amb cap llindar.

Com es pot veure a la figura 8.4, la xarxa no és capaç de generalitzar del tot, és a dir, no és capaç de diferenciar bé els dos casos. Independentment de si s'equivoca o no, la majoria de cops dona unes prediccions properes als extrems. O sigui, gran part dels clients que rebutja, ho fa amb prediccions properes al 0, i el mateix passa amb la majoria dels impostors que accepta, però amb prediccions properes a 1.

Per mesurar la fiabilitat de la xarxa neuronal siamesa, he utilitzat les següents mètriques.

- *False Acceptance Rate* (FAR). Mesura del percentatge de persones sense autorització que han estat acceptades per error.
- *False Recognition Rate* (FRR). Mesura del percentatge de persones autoritzades que han estat rebutjades per error.

Un cop s'han obtingut els resultats, es calculen els FARs i els FRRs d'aquests, utilitzant un conjunt de 200 llindars diferents, compresos entre 0 i 1, per decidir a quin cas pertanyen. O sigui, se simula, per a cada llindar, el que passaria si els clients i impostors intentessin accedir al sistema configurat amb aquest llindar, i se'n calcula el FRR i el FAR. Com a resultat d'això, he obtingut el gràfic de la figura 8.5.

A banda de mesurar la fiabilitat de la xarxa, aquest procés també ha servit per trobar el llindar òptim, és a dir, amb el que obtenim millors resultats. El llindar òptim obtingut és de 0,5, el qual s'utilitza a la demostració, i correspon al punt vermell de la figura 8.5, on FAR és igual a FRR. Aquest punt correspon a l'error òptim (ERR), que en aquest cas és del 29,77%.

[8]Sistema automatitzat de reconeixement humà basat en les característiques físiques d'aquests (empremta, veu, cara, iris, etc.).

[9]El procés s'ha realitzat a través del codi `neuralnetwork/score_generator.cc` del repositori.

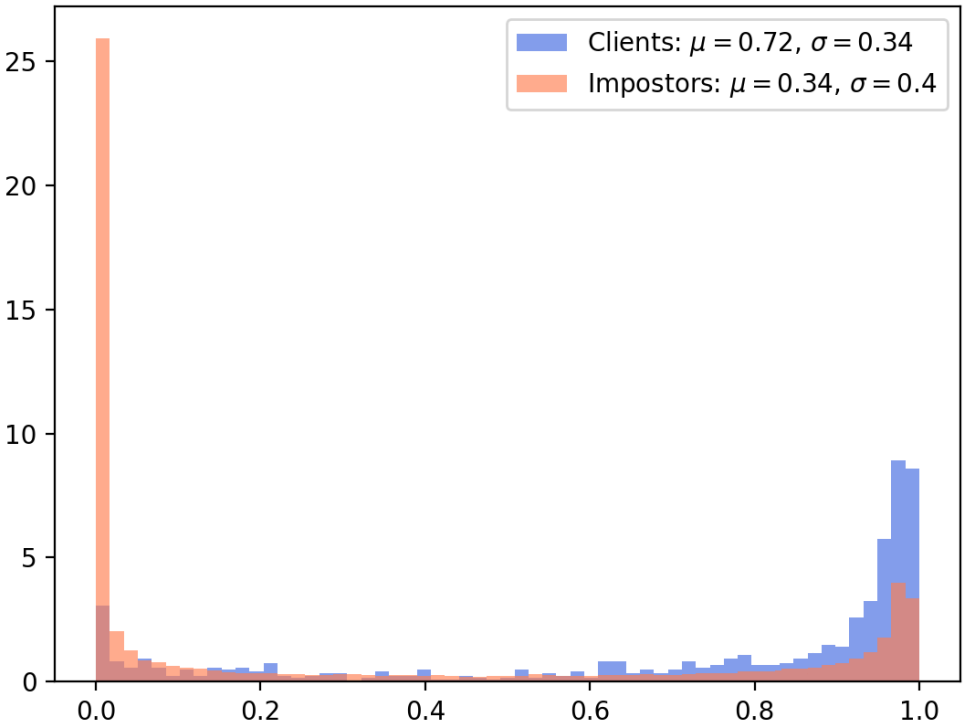


Figura 8.4: Histograma dels resultats.

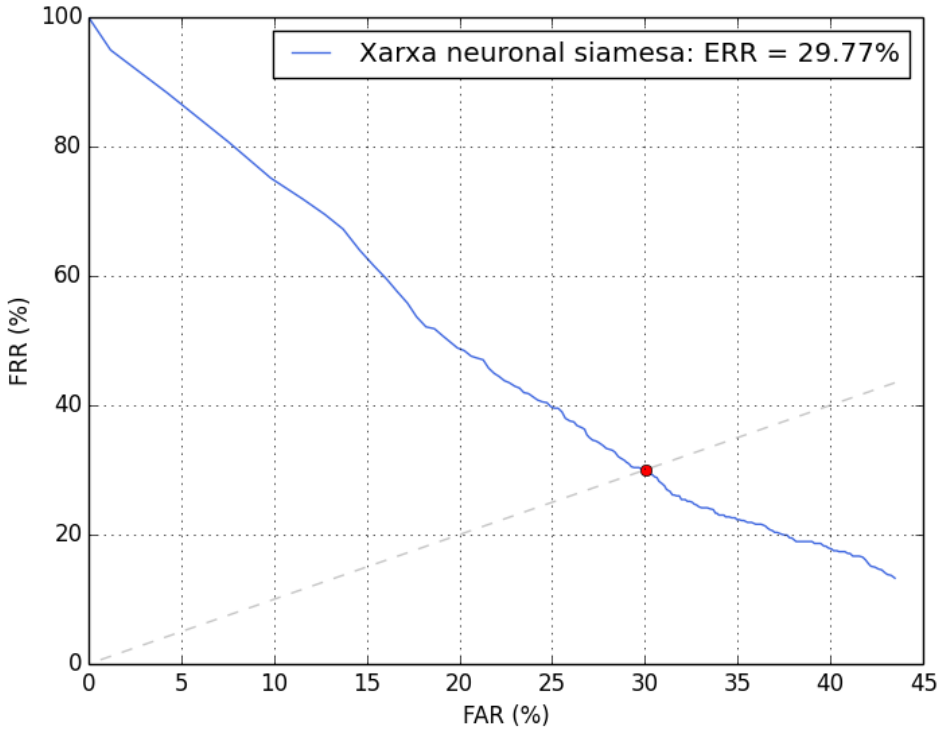


Figura 8.5: Corba DET (*detection error tradeoff*).

8.2 Demostració

La demostració[10] és una interfície gràfica programada en HTML, CSS i JavaScript, utilitzant el *framework* Electron per poder generar una aplicació amb aquest codi.

Aquesta demostració està dissenyada per poder seleccionar 2 locutors i un fitxer d'àudio de cadascun d'ells, per escoltar aquests fitxers i comparar-los. Està pensada per poder reproduir només un dels dos fitxers a la vegada, i només es podrà reproduir quan estigui seleccionat. A més a més, per poder executar la comparació cal que s'hagin triat els dos fitxers i que la xarxa s'hagi carregat correctament.

Els fitxers d'àudio utilitzats formen part de les dades de prova, és a dir, no hi ha cap locutor que la xarxa hagi vist abans. Com ja he comentat a l'apartat 8.1.1, els fitxers són en anglès. Per aquest motiu, i perquè és l'idioma més utilitzat, he realitzat la demostració en anglès.

A continuació es mostren els elements que es troben a la interfície i una breu explicació de la seva utilitat.

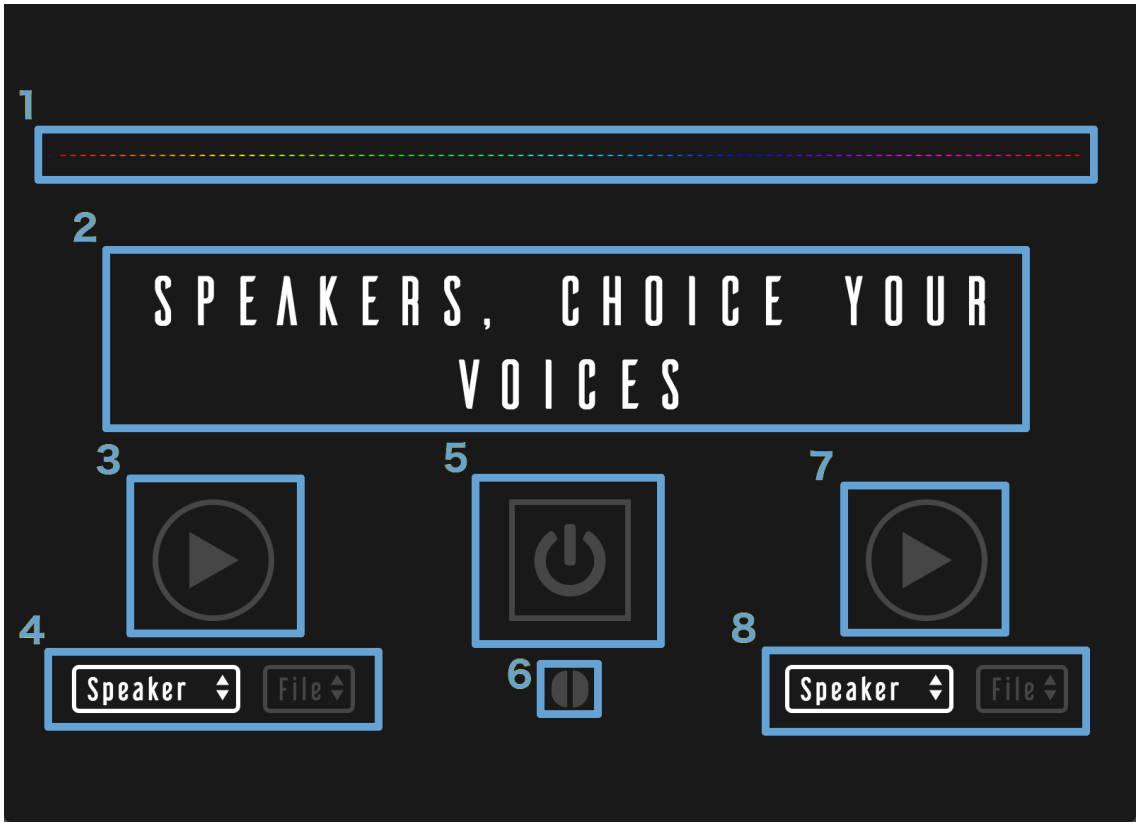


Figura 8.6: Captura de la interfície només obrir-la (inici).

1. Representació gràfica dels fitxers d'àudio. Només es mou quan es reprodueix un d'ells.

[10]La demostració està formada pel codi que conté la carpeta `ui/` del repositori.

2. Guia per l'usuari. Dona ordres a l'usuari perquè sàpiga el que ha d'anar fent, i mostra el resultat final.
3. Botó de reproducció del fitxer del primer locutor. Només es pot pitjar si se selecciona locutor i fitxer.
4. Seleccionadors del primer locutor i del seu fitxer. Només deixarà escollir el fitxer un cop s'hagi triat el locutor.
5. Botó d'execució. Només es pot pitjar si estan seleccionats els dos locutors i els dos fitxers, i si la xarxa s'ha carregat correcta i completament.
6. Indicador de fitxers. Mostra si s'ha carregat el primer (semicercle esquerre) i el segon fitxer (semicercle dret) quan s'acoloreix.
7. Botó de reproducció del fitxer del segon locutor. Només es pot pitjar si se selecciona locutor i fitxer.
8. Seleccionadors del segon locutor i del seu fitxer. Només deixarà escollir el fitxer un cop s'hagi triat el locutor.

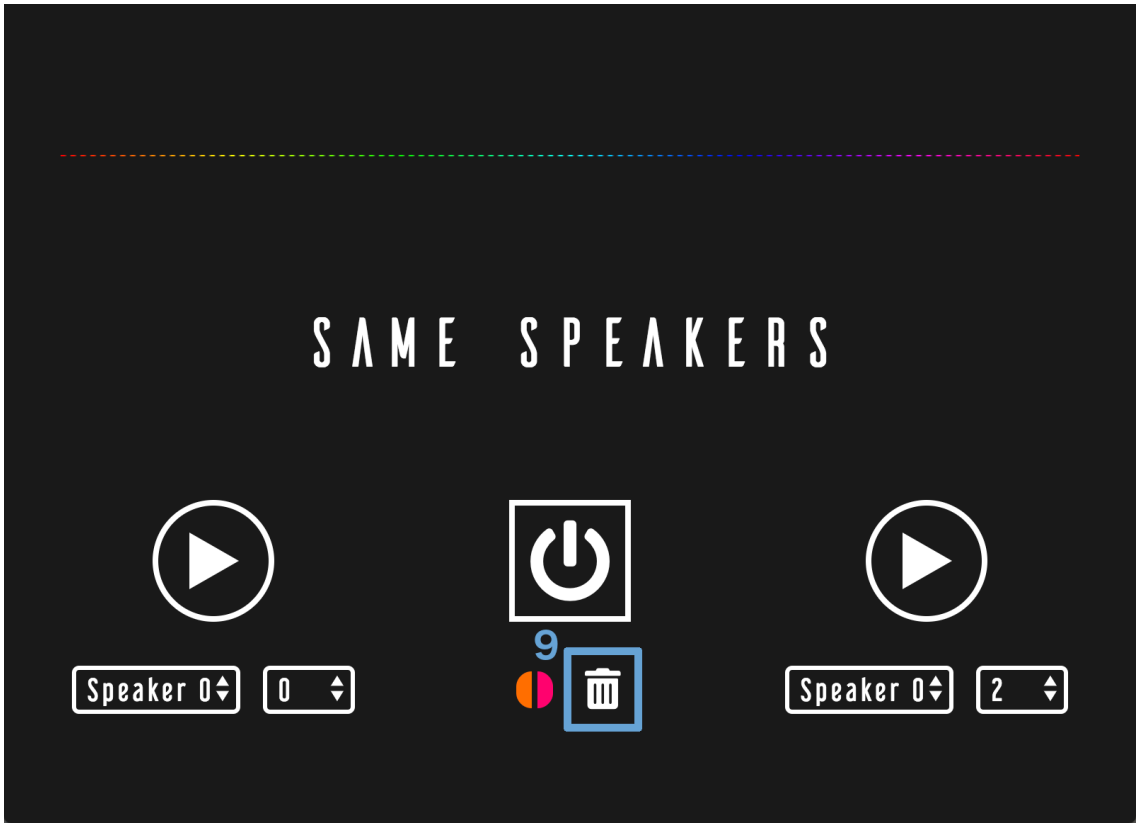


Figura 8.7: Captura de la interfície després d'executar la comparació.

9. Botó de *reset*. Un cop s'hagi executat la comparació, apareix aquest botó. Si es prem, la demostració torna a l'estat inicial, com si s'acabés d'obrir (on aquest botó no està).

Capítol 9

Sostenibilitat i compromís social

9.1 Autoavaluació

Durant la meva estança en aquest grau, només he tractat la sostenibilitat en les competències transversals, xerrades, i a l'assignatura d'Arquitectura del PC. Gràcies a aquesta, els meus coneixements sobre la sostenibilitat han augmentat considerablement.

Actualment, aquests coneixements són sobretot teòrics i generals, és a dir, conec molts dels problemes actuals (e.g. *e-waste*), nous moviments que poden aportar solucions (e.g. economia circular), els principis deontològics, eines de treball col·laboratiu, accessibilitat, ergonomia, etc. A l'hora de començar un projecte puc tenir en compte tot el que sé per intentar realitzar-lo de la millor manera possible, tenint sempre presents els impactes ambiental, econòmic i social del projecte.

Tot i això, encara no he obtingut els coneixements necessaris per poder realitzar-ho tenint en compte tots els aspectes que abasta la sostenibilitat, detectant els problemes i solucionant-los, valorant els impactes del projecte a la societat i a la sostenibilitat del planeta, mesurant amb els indicadors adequats la seva contribució a la millora de la societat, etc. A més a més, com no he participat en cap projecte real, amb conseqüències reals (més enllà d'una nota a l'expedient), no he pogut aplicar cap tipus de coneixement sobre el tema.

9.2 Impacte Ambiental

L'impacte ambiental que podia provocar aquest projecte era mínim. Estava previst que es desenvolupés tot amb un sol ordinador (el del propi desenvolupador). Totes les eines *software* haurien estat en línia, disponibles, tot i no ser utilitzades en el treball. A més a més, l'energia consumida en el seu desenvolupament havia de ser només la utilitzada per aquest ordinador. A part, el transport públic i la fibra òptica s'anaven a seguir utilitzant en qualsevol cas. Per tant, no creia possible la minimització de l'impacte del treball sense la reducció del seu temps de desenvolupament, cosa que hagués afectat al producte final o hagués augmentat els costos en recursos humans, sense la garantia d'una reducció signi-

ficativa d'aquest impacte. Finalment, el projecte ha tingut l'impacte ambiental esperat i no s'ha pogut realitzar una minimització de l'impacte. L'únic canvi, respecte a la previsió inicial, ha estat l'energia utilitzada per *Calcula*.

En el cas de tornar a realitzar el projecte, crec que podria utilitzar menys recursos reduint el temps de desenvolupament. He après moltes coses durant aquest temps, coses que no hauria de tornar a aprendre. No malgastaria un altre cop la gran quantitat de recursos i de temps que he destinat a l'aprenentatge. Ha estat una despesa necessària, però, si tornés a desenvolupar el projecte, no hauria de produir-se.

Com es va explicar a l'apartat 1.2, actualment hi ha dues empreses que es dediquen al desenvolupament de *software* de reconeixement de locutors. A diferència d'elles, aquest projecte és molt més petit i, per tant, la seva realització i manteniment suposaran un menor impacte ambiental.

Actualment, el recurs que s'utilitzarà durant la vida útil del projecte és l'energia que gastin els ordinadors dels usuaris del *software* mentre l'utilitzen, la qual es preveu que serà poca. En el cas que algú ampliï el meu treball, incloent-lo al *TextServer*, s'haurien de tenir en compte els recursos que utilitzi el servidor.

L'ús del *software* no reduirà el consum d'altres recursos, almenys no en la seva forma actual. L'únic possible escenari on el *software* és capaç de fer-ho, és el d'una persona comparant fitxers d'àudio de persones parlant, a mà.

L'únic risc d'augment de la petjada ecològica del projecte és l'augment del nombre d'usuaris del *software*.

9.3 Impacte Econòmic

Tots els aspectes econòmics del projecte han estat tractats al capítol 6. Allà s'especifiquen el pressupost inicial i el cost real de la realització del treball. Com ja he comentat al capítol, el cost final supera 765 € el pressupost inicial. Això ha estat causat per l'increment d'hores de desenvolupament (70 h més de les esperades), i la inclusió en el pressupost de la xarxa d'ordinadors *Calcula*.

Encara que no ho he tingut en compte en el cost final, aquest projecte podria augmentar el seu cost si s'acaba incloent a la plataforma *TextServer*, tant en recursos humans com materials.

A part dels costos, aquest projecte tindrà un altre impacte en l'economia. A diferència de les dues opcions més importants del mercat (*software* tancat i de pagament), aquest treball permetrà a petits (i grans) desenvolupadors utilitzar el mòdul de verificació de locutors en el seu propi projecte, de forma gratuïta. A més a més, ajudarà al centre TALP a donar més visibilitat a la seva recerca, i té potencial per enriquir el *TextServer*, tal com s'ha comentat a l'apartat 1.3.3.

No existeix cap escenari on el projecte no sigui econòmicament viable.

9.4 Impacte Social

En l'àmbit personal, aquest treball m'ha fet créixer com a enginyer informàtic, sobretot en el camp en el qual vull enfocar la meva vida professional. He après molts conceptes nous sobre un tema punter i interessant com ho és el *deep learning* i sobre la caracterització de locutors, realitzant un projecte que em va cridar l'atenció des del primer dia que vaig saber d'ell. A més a més, m'ha servit com a primera presa de contacte amb un projecte real, on he hagut de complir unes dates, redactar una documentació formal, realitzar una presentació davant d'un tribunal, etc. Coses que, a la universitat, els alumnes no l'hi donem importància, però a la vida professional es valoren moltíssim.

Com ja s'ha comentat en l'apartat anterior, aquest projecte beneficiarà a tot desenvolupador que necessiti un *software* com el mòdul de verificació de locutors, ja que és gratuït i de lliure disposició. De moment ho és a través del repositori de *GitHub*, més endavant potser a través del *TextServer*. A més a més, la demostració permet comparar dos fitxers d'àudio per determinar si en aquests parla el mateix locutor o no, i també és accessible per tothom.

Tot i que el *software* desenvolupat per *AGNITO* i *Nuance* no és comparable al del projecte, sí que ha existit una necessitat real per realitzar-lo: incrementar la visibilitat de la recerca de TALP i la seva potencial incorporació a la plataforma *TextServer* per enriquir-la.

Per acabar, comentar que el treball no és perjudicial per ningú. És un projecte petit, no competeix amb cap altre *software*, i no està pensat per fer res més enllà de comparar fitxers d'àudio.

Capítol 10

Conclusions

Encara que hagi tingut problemes al llarg del treball, que les planificacions temporals i econòmiques no hagin estat prou precises, i que no hagi complert els objectius secundaris, el resultat final ha estat satisfactori. He aconseguit crear un *software* amb interfície gràfica, capaç de comparar dues veus (i reproduir-les) amb un error òptim (ERR) del 29,77%.

Aquest error és una de les limitacions del treball. S'hauria pogut incrementar amb un millor preprocessament de les dades i amb un millor entrenament de la xarxa neuronal siamesa. L'altra limitació ha estat la naturalesa inherent dels fitxers d'àudio. En ser el resultat de converses telefòniques, tenen el so característic d'aquestes, la qual cosa limita l'ús del *software* a fitxers de converses telefòniques. D'altra forma, l'error encara seria més alt.

Per projectes futurs, es poden realitzar moltes millores al meu Treball de Fi de Grau. Es pot incloure al *TextServer*, complint així amb els objectius secundaris d'aquest treball. Es poden realitzar canvis en el funcionament de la demostració, permetent gravar la veu de l'usuari en lloc de fer-li sel·leccionar una. A part d'aquestes millores, podria utilitzar-se de base d'aplicacions que necessitin comparar fitxers d'àudio, com per exemple per desenvolupar un sistema biomètric de verificació de locutor (com les proves que s'han realitzat a l'apartat 8.1.3). Fins i tot podria ser un punt de partida per desenvolupar un identificador de locutors.

Bibliografia

[1] Douglass A. Reynolds, Richard C. Rose (1995). *Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models*. URL: http://www.cs.toronto.edu/~frank/csc401/readings/ReynoldsRose.pdf [Última consulta: 6 març de 2018]

[2] Douglass A. Reynolds (1994). *Speaker identification and verification using Gaussian mixture speaker models*. URL: http://visgraph.cs.ust.hk/biometrics/Papers/Voice/reynolds_spcomm1995.pdf [Última consulta: 6 març de 2018]

[3] Ladan Baghai-Ravary, Steve W. Beet. *Automatic Speech Signal Analysis for Clinical Diagnosis and Assessment of Speech Disorders*. ed. Springer Science & Business Media, 2012. ISBN 1461445744, 9781461445746.

[4] Wikipedia, the free encyclopedia. *Deep Learning*. URL: https://en.wikipedia.org/wiki/Deep_learning [Última consulta: 6 de març de 2018]

[5] *Textserver*. URL: http://textserver.cs.upc.edu/textserver [Última consulta: 6 de març de 2018]

[6] *TALP :: Language and Speech Technologies and Applications - UPC*. URL: http://www.talp.upc.edu/ [Última consulta: 6 de març de 2018]

[7] Margaret Rouse (2005). *Prototyping Model*. URL: http://searchcio.techtarget.com/definition/Prototyping-Model [Última consulta: 6 març de 2018]

[8] ISTQB Exam Certification. *What is Prototype model - advantages, disadvantages and when to use it?*. URL: http://istqbexamcertification.com/what-is-prototype-model-advantages-disadvantages-and-when-to-use-it [Última consulta: 6 març de 2018]

[9] *Git*. URL: https://git-scm.com [Última consulta: 6 març de 2018]

[10] *GitHub*. URL: https://github.com/ [Última consulta: 6 març de 2018]

[11] Sadaoki Furui (2005). *50 Years of Progress in Speech and Speaker Recognition Research*. URL: https://www.tci-thaijo.org/index.php/ecticit/article/view/51834/42958 [Última consulta: 7 març de 2018]

[12] P. Kenny, V. Gupta, T. Stafylakis, P. Ouellet and J. Alam (2014). *Deep Neural Networks for extracting Baum-Welch statistics for Speaker Recognition*. URL: http://www.isca-speech.org/archive/odyssey_2014/pdfs/28.pdf [Última consulta: 7 març de 2018]

[13] Alex Perala (2016). *Nuance Communications Acquires Agnitio*. URL: https://findbiometrics.com/nuance-acquires-agnitio-311101 [Última consulta: 7 març de 2018]

[14] AGNITIO. *About Us*. URL: http://www.agnitio-corp.com/company/aboutus [Última consulta: 7 març de 2018]

[15] Nuance. *Speech Recognition Solutions*. URL: https://www.nuance.com/mobile/speech-recognition-solutions.html [Última consulta: 7 març de 2018]

[16] Apple. *MacBook Pro - Especificaciones*. URL: https://www.apple.com/es/macbook-pro/specs/ [Última consulta: 14 de març de 2018]

[17] Apple. *macOS High Sierra*. URL: https://www.apple.com/es/macos/high-sierra/ [Última consulta: 14 de març de 2018]

[18] Canonical Group Ltd. *The leading operating system for PCs, loT devices, servers and the cloud — Ubuntu*. URL: https://www.ubuntu.com/ [Última consulta: 15 juny de 2018]

[19] Microsoft. *Virtual Studio Code*. URL: https://code.visualstudio.com/ [Última consulta: 14 març de 2018]

[20] Electron. *Electron — Build cross platform desktop apps with JavaScript, HTML, and CSS* . URL: https://electronjs.org/ [Última consulta: 17 juny de 2018]

[21] *Git*. URL: https://git-scm.com [Última consulta: 14 març de 2018]

[22] *GitHub*. URL: https://github.com/ [Última consulta: 14 març de 2018]

[23] LaTeX. URL: https://www.latex-project.org/ [Última consulta: 14 març de 2018]

[24] *GanttProject*. URL: http://www.ganttproject.biz/ [Última consulta: 14 març de 2018]

[25] SoftCatalà. *Corrector ortogràfic i gramatical*. URL: https://www.softcatala.org/corrector/ [Última consulta: 14 març de 2018]

[26] Graham Neubig and Chris Dyer and Yoav Goldberg and Austin Matthews and Waleed Ammar and Antonios Anastasopoulos and Miguel Ballesteros and David Chiang and Daniel Clothiaux and Trevor Cohn and Kevin Duh and Manaal Faruqui and Cynthia Gan and Dan Garrette and Yangfeng Ji and Lingpeng Kong and Adhiguna Kuncoro and Gaurav Kumar and Chaitanya Malaviya and Paul Michel and Yusuke Oda and Matthew Richardson and Naomi Saphra and Swabha Swayamdipta and Pengcheng Yin. *Dynet: The Dynamic Neural Network Toolkit*. URL: http://dynet.io/ [Última consulta: 27 maig de 2018]

[27] GitHub. *MNIST DyNet example at GitHub*. URL: https://github.com/clab/dynet/tree/master/examples/mnist [Última consulta: 17 juny de 2018]

[28] Google. *Google Cloud Platform Pricing Calculator*. URL: https://cloud.google.com/products/calculator [Última consulta: 23 juny de 2018]

[29] NVIDIA. *Tarjeta gráfica NVIDIA TITAN Xp*. URL: http://www.nvidia.es/graphics-cards/geforce/pascal/titan-xp/ [Última consulta: 23 juny de 2018]

[30] Omid Ghahabi, Javier Hernando. *Restricted Boltzmann machines for vector representation of speech in speaker recognition*.

[31] Douglass A. Reynolds. *Universal Background Models*. URL: https://www.ll.mit.edu/mission/cybersec/publications/publication-files/full_papers/0802_Reynolds_Biometrics_UBM.pdf [Última consulta: 23 juny de 2018]